Software Requirements Specification

for

Cyber Range

Version 1.0 approved

Prepared by
xxx
xxx
xxx
xxx

Sepuluh Nopember Institute of Technology

2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to build a simulated environment for cybersecurity training, testing, and skill development.

1.2 Document Conventions

This document follows a standardized format that includes the use of fonts and special emphasis to facilitate understanding and navigation as per the IEEE standard template for System Requirement Specification Documents.

1.2.1 Heading Styles

Different heading levels are utilized to structure the document, such as Introduction and System Overview.

1.2.2 Font Styles

Bold is employed for section headings and to emphasize essential terms, while italic is used for variable names or to highlight certain points.

1.2.3 Lists

Information is organized using bullet points or numbered lists, facilitating clear presentation of requirements or related details.

1.2.4 Tables

Tables are employed for organized presentation, particularly for listing requirements, use cases, or data.

1.2.5 Emphasis

Font styles are applied to emphasize key points or terms within the document.

1.2.6 Page Numbers and Headers/Footers:

Headers or footers include page numbers and the document title for easy navigation.

1.2.7 Acronyms and Abbreviations

Acronyms and abbreviations are defined upon first usage to ensure clarity throughout the document.

1.2.8 Use of Diagrams

Visual representations like use case diagram are included to enhance understanding of system components and processes.

1.2.9 Revision History

A section for version control is integrated to track document revisions.

1.2.10 Annotations

Comments or annotations may be included where necessary to provide additional context.

1.3 Intended Audience and Reading Suggestions

This project serves as a prototype for the Cyber Range Platform, limited to use within the academic premises and for internal organizational purposes. It has been developed under the guidance of college professors. The project is beneficial for students who are eager to gain practical experience in cybersecurity and enhance the internal organization's capabilities.

To delve deeper into the foundational principles of the system, please refer to the *references* section, which serves as the primary source for most aspects of the project.

1.4 Product Scope

The purpose of the Cyber Range Platform is to provide realistic threat simulations, interactive exercises, research support, and custom scenarios for effective cybersecurity training and readiness. The system operates on the concept of Infrastructure as Code (IaC), utilizing it to generate virtual machines with customized network topologies. Additionally, a relational database is employed to store user information, session data, task details, reports, and environmental configurations.

A pivotal component of this platform is the Hypervisor Server, dedicated to hosting multiple sandboxed virtual environments. These environments play a pivotal role in simulating real-life scenarios, contributing to a comprehensive and hands-on learning experience. The overarching goal is to provide the most realistic and optimal approach for individuals seeking to immerse themselves in the realm of cybersecurity education.

1.5 References

- *HackTheBox*
- *TryHackMe*
- *Kypo*
- *Cyber Defenders*

2. Overall Description

2.1 Product Perspective

A simulated environment for cybersecurity training, testing, and skill development. Cyber Range platform is essential for user-friendly, personalized cybersecurity training and awareness. The platform provides realistic threat simulations, interactive exercises, research support, and custom scenarios for effective cybersecurity training and readiness.

The Cyber Range platform serves as a pivotal resource for fostering comprehensive cybersecurity education. Tailored for both university students and professionals, it is designed to deliver practical training that imparts invaluable hands-on experience. For university education, the platform offers a dynamic setting where students can engage in realistic threat simulations, interactive exercises, and customized scenarios, cultivating a deeper understanding of cybersecurity principles. Meanwhile, in the realm of professional training, the Cyber Range equips cybersecurity experts with the tools to explore current threats and develop effective prevention strategies. Going beyond traditional approaches, the platform embraces training gamification, integrating game-like elements and mechanics into the learning environment. This not only makes the training experience more enjoyable but also enhances engagement, motivating participants to actively participate in honing their cybersecurity skills and awareness. Ultimately, the Cyber Range platform stands as a multifaceted solution, seamlessly integrating into educational and professional contexts to elevate cybersecurity readiness.

2.2 Product Functions

2.2.1 Register new user
2.2.2 Update profile
2.2.3 Delete user
2.2.4 User Login
2.2.5 Define sandbox environment
2.2.6 Define training tasks
2.2.7 Define training session
2.2.8 Play training
2.2.9 Generate training report
2.2.10 Give feedback
2.2.11 Access sandbox environment
2.2.12 Reset sandbox
2.2.13 Stop sandbox
2.2.14 View training report
2.2.15 View feedback

2.3 User Classes and Characteristics

2.3.1 Administrator

This role is tasked with the responsibility of effectively overseeing and managing user accounts within the system.

2.3.2 Training User

As the primary participant in the training process, the Training User assumes a central role. They possess the capability to initiate an environment, engage with it, and successfully complete assigned tasks. Additionally, Training Users have the ability to monitor and assess their own performance throughout the training sessions.

2.3.3 Training Maker

The Training Maker holds the crucial role of crafting scenarios that the Training User will navigate, along with defining the objectives to be achieved within these scenarios. Moreover, the Training Maker is responsible for validating the suitability of sessions for their designated environments, ensuring the effectiveness and relevance of the training experience.

2.4 Operating Environment

2.4.1 Architecture

The system follows a client/server model, prioritizing efficient communication and data exchange between the two entities.

2.4.2 Operating System

Linux is the chosen operating system, selected for its stability, security features, and flexibility.

2.4.3 Hypervisor Server

Proxmox serves as the designated hypervisor, providing virtualization capabilities and resource management for optimal performance.

2.4.4 Infrastructure as Code (IaC)

Terraform and Ansible are employed as the primary tools for defining and provisioning the infrastructure, ensuring consistent and reproducible deployment automation.

2.4.5 Database

The system relies on a SQL database, chosen for its relational data management capabilities, scalability, and robust transaction support.

2.4.6 Application Server

The application is built on a dual-stack foundation, utilizing NextJS for the frontend and Golang for the backend. This combination offers a dynamic and responsive user interface, coupled with a robust and scalable server-side architecture.

2.4.7 Architectural Flow

Frontend → Backend → Terraform → Hypervisor Server

2.5 Design and Implementation Constraints

This section outlines the specific limitations and constraints that shape the design and implementation of the project. Understanding these constraints is crucial for stakeholders and developers to align expectations and make informed decisions throughout the development process.

2.5.1 User Base Limitation

The project is exclusively tailored for computer users, and mobile usage is not supported. This decision is driven by a focus on providing a robust and optimized experience for desktop or laptop users. Any considerations for mobile use are beyond the scope of this project.

2.5.2 Virtual Machine Configuration Constraints

Upon the generation and launch of virtual machines (VMs) using Terraform, the configuration is dependent on templates provided by the Proxmox hypervisor server. Devs and Admins are granted autonomy to choose the appropriate template during the generation of Terraform configurations. It is crucial to note that the selection of templates is not integrated into the transactional CRUD (Create, Read, Update, Delete) operations of the application server. Requests for new templates must be handled manually by accessing the hypervisor server directly, ensuring that the development team has the flexibility to adapt to evolving requirements.

2.5.3 Limited Functionality for Training Tasks, Sessions, and Sandboxes

This project intentionally excludes the provision of edit and delete functionality for training tasks, sessions, and sandboxes. This design choice is motivated by synchronization considerations and a commitment to preserving a seamless user experience. If there arises a need for edit and delete functionality despite these constraints, the resolution requires manual configuration accessed directly on the server. This ensures that any modifications to these functionalities are approached with careful consideration and are executed in a controlled manner.

It's essential to recognize that these constraints are fundamental to the project's design philosophy and serve as guidelines for both development and usage scenarios. Stakeholders and developers should be mindful of these limitations when planning and interacting with the system.

2.6 User Documentation

The Cyber Range Platform is designed exclusively for use within the academic premises and internal organizational purposes. Given the nature of its use, user documentation and tutorials will be conducted by instructors and/or administrators directly before users engage with the platform. These tutorials are designed to ensure that users, both from academic and non-academic backgrounds, receive personalized guidance on utilizing the platform effectively.

2.7 Assumptions and Dependencies

This section outlines the key assumptions and dependencies that underpin the successful development, deployment, and operation of the system.

2.7.1 Infrastructure Capacity

The production environment possesses sufficient infrastructure capability to host the project effectively. Given the anticipated high demand for virtual machines (VMs) and related resources, it is crucial that the hosting environment can scale to meet the dynamic requirements of the project.

2.7.2 Resource Stability

The assumption is made that the network connection between client and server components remains consistently stable. As the system involves extensive synchronization and frequent socket openings, it is imperative that the server has ample resources to manage these processes efficiently. Adequate computational resources, including CPU, memory, and network bandwidth, are essential for optimal performance.

2.7.3 VPN Configuration

A Virtual Private Network (VPN) is configured to facilitate secure user access to VMs through the internal network. This assumption ensures that users can securely connect to the system, maintaining data integrity and confidentiality during interactions with virtual environments. The system's functionality and security rely on the proper configuration and ongoing maintenance of the VPN infrastructure.

2.7.4 Infrastructure as Code (IaC) Tools

Terraform and Ansible serves as the primary tool for configuring the Infrastructure within the Virtual Environment. It is essential to note that the usage of the terraform-proxmox provider may be subject to depreciation, considering it is not officially supported. The project's functionality could be impacted if this assumption proves inaccurate or if there are changes to the provider's status.

2.7.5 Network Stability

The successful execution of the system relies on a stable network connection between client and server components. Assumptions include minimal latency, adequate bandwidth, and reliable data transmission. Any deviations from these assumptions may affect the real-time synchronization and performance of the system.

2.7.6 Proxmox Hypervisor Updates

Routine updates and maintenance of the Proxmox hypervisor are assumed not to introduce incompatible changes that may impact system functionality. Developers must monitor the Proxmox updates and promptly address any unforeseen issues that could arise due to changes in the hypervisor's status.

2.7.7 Third-Party Library Compatibility

Dependencies on external libraries, frameworks, or tools not explicitly mentioned may exist. Assumptions regarding the compatibility and stability of these components should be considered, as changes or issues with third-party elements can affect system performance. Regular monitoring and updates may be required to address any changes in third-party library compatibility.

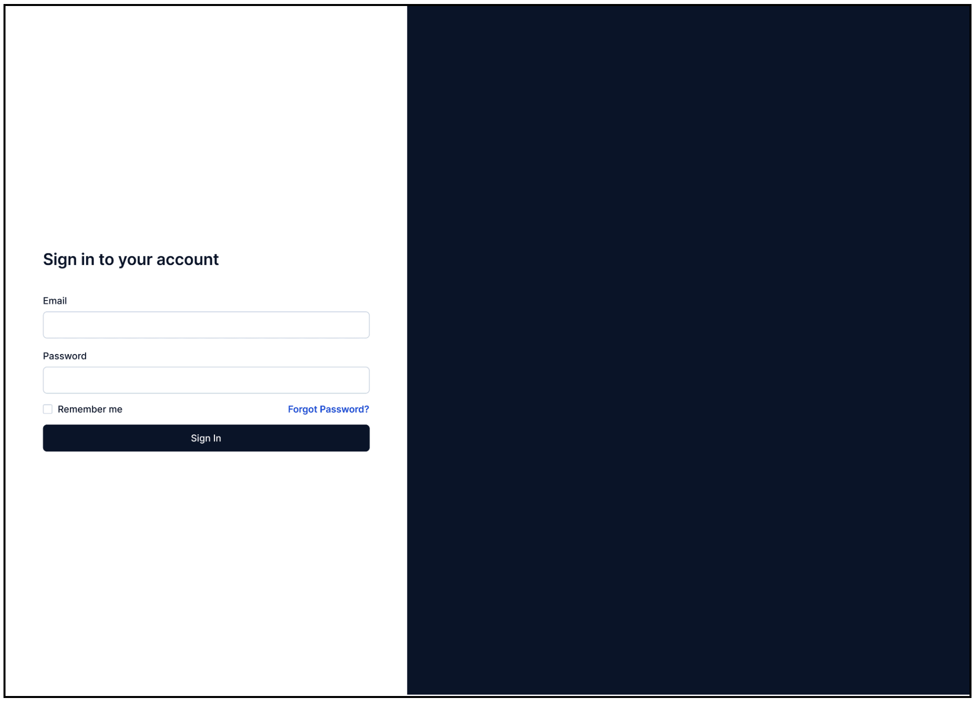
2.7.8 Security Patching

The assumption is that the operating system, database, and other system components will receive timely security patches and updates. Failing to address security updates promptly may expose the system to vulnerabilities, potentially compromising the overall security of the infrastructure. Regular maintenance and monitoring of security patches are crucial for maintaining system integrity.

3. External Interface Requirements

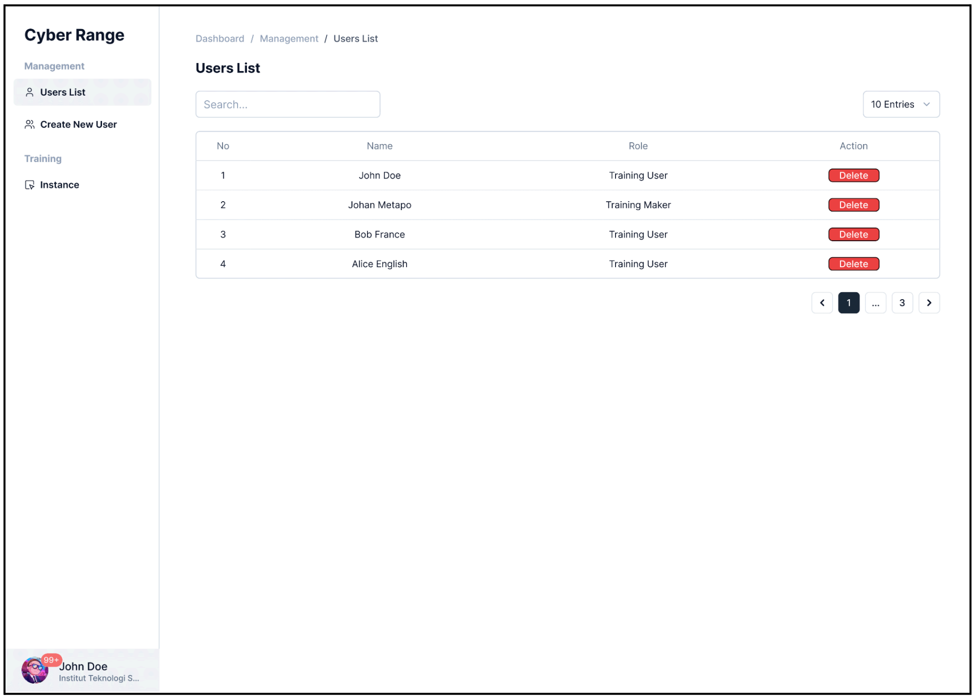
3.1 User Interfaces

3.1.1 Authentication



3.1 Authentication

3.1.2 Admin

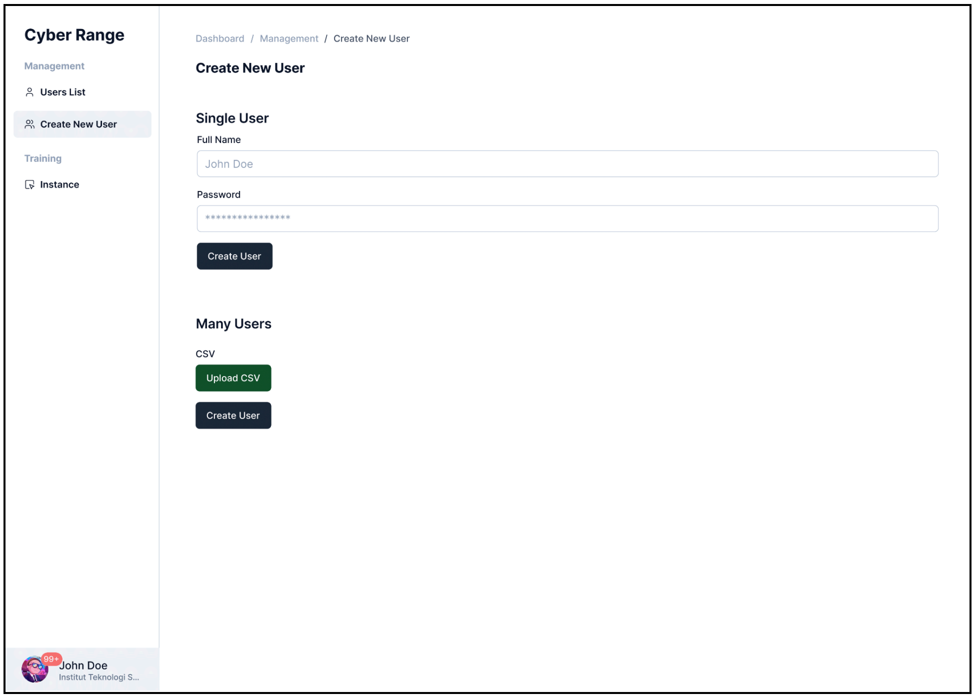


Dashboard / Management / Users List

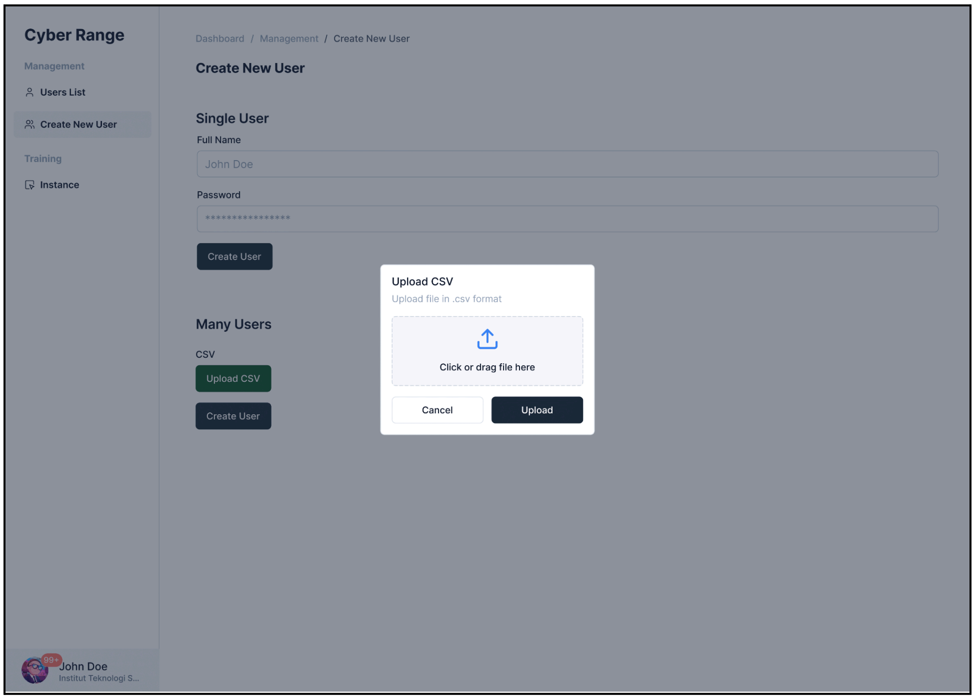
Users List

No	Name	Role	Action
1	John Doe	Training User	Delete
2	Johan Metapo	Training Maker	Delete
3	Bob France	Training User	Delete
4	Alice English	Training User	Delete

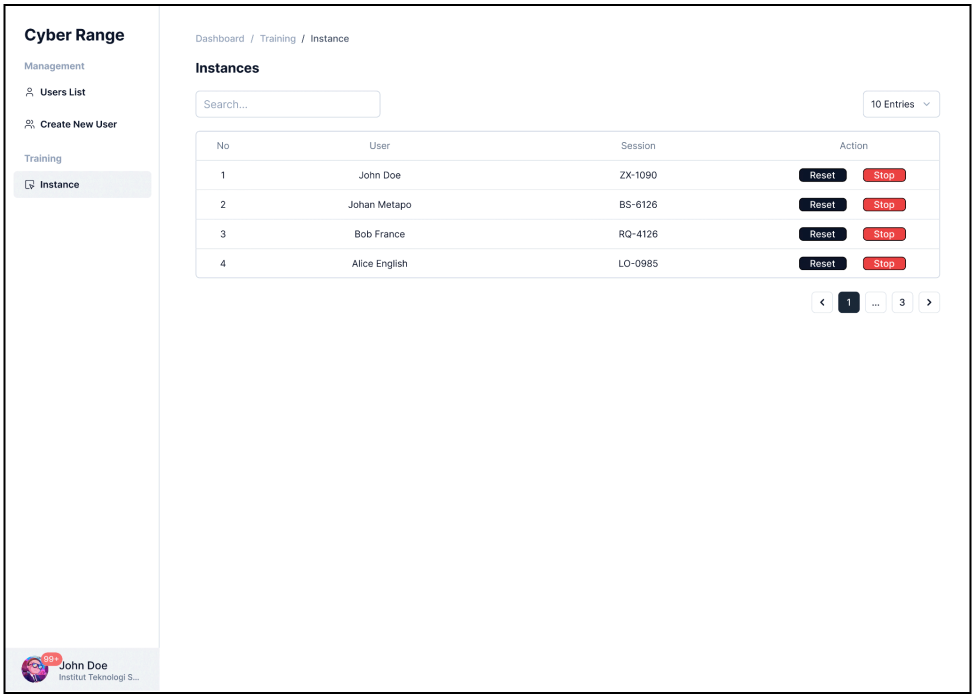
3.2 Admin User List



3.3 Admin Create New User



3.4 CSV Upload



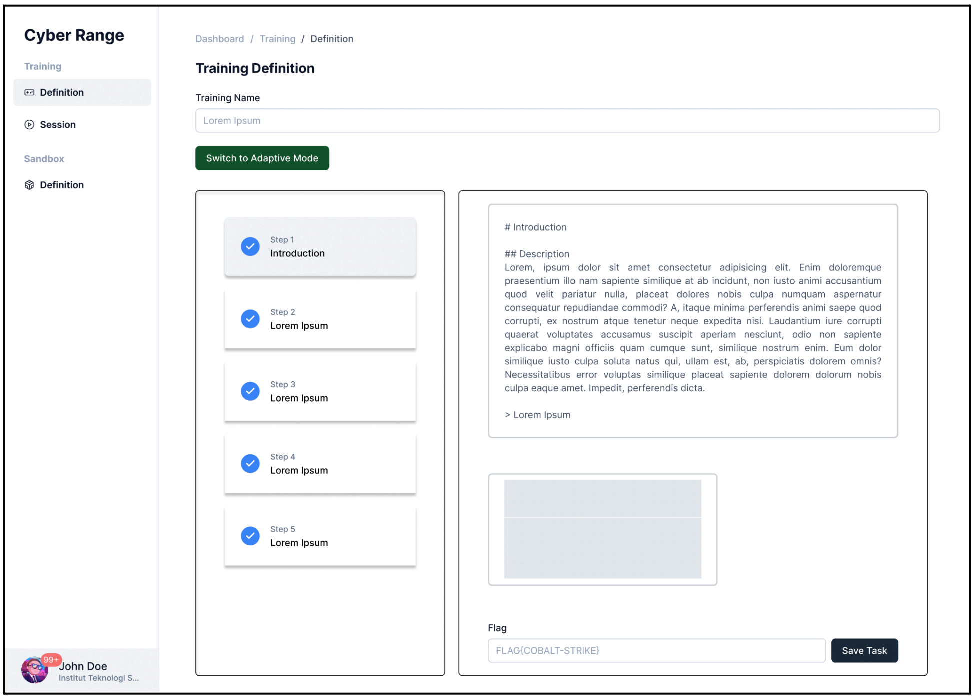
Dashboard / Training / Instance

Instances

No	User	Session	Action
1	John Doe	ZX-1090	Reset Stop
2	Johan Metapo	BS-6126	Reset Stop
3	Bob France	RQ-4126	Reset Stop
4	Alice English	LO-0985	Reset Stop

3.5 Instance

3.1.3 Training Maker



Dashboard / Training / Definition

Training Definition

Training Name: Lorem Ipsum

- Step 1 Introduction
- Step 2 Lorem Ipsum
- Step 3 Lorem Ipsum
- Step 4 Lorem Ipsum
- Step 5 Lorem Ipsum

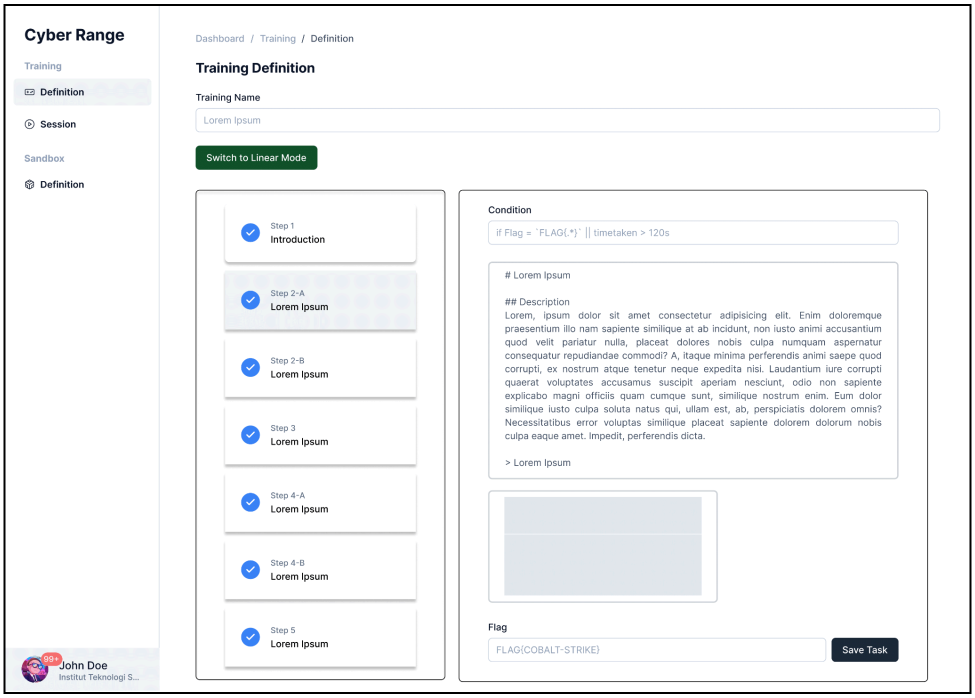
```
# Introduction

## Description
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Enim doloremque praesentium illo nam sapiente similique at ab incidunt, non iusto animi accusantium quod velit pariatur nulla, placeat dolores nobis culpa numquam aspernatur consequatur repudiandae commodi? A, itaque minima perferendis animi saepe quod corrupti, ex nostrum atque tenetur neque expedita nisi. Laudantium iure corrupti quaerat voluptates accusamus suscipit aperiam nesciunt, odio non sapiente explicabo magni officiis quam cumque sunt, similique nostrum enim. Eum dolor similique iusto culpa soluta natus qui, ullam est, ab, perspiciatis dolorem omnis? Necessitatibus error voluptas similique placeat sapiente dolorem dolorum nobis culpa eaque amet. Impedit, perferendis dicta.

> Lorem Ipsum
```

Flag: FLAG{COBALT-STRIKE}

3.6 Linear Training Definition



Training Definition

Training Name: Lorem Ipsum

Condition: if Flag = `FLAG{.*}` || timetaken > 120s

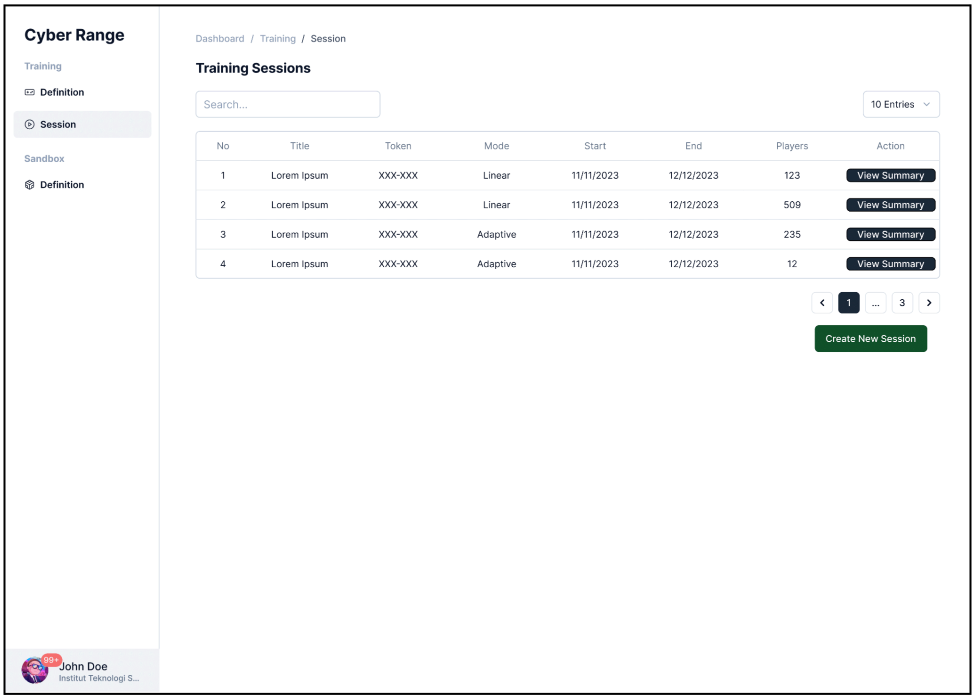
```
# Lorem Ipsum

## Description
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Enim doloremque praesentium illo nam sapiente similique at ab incidunt, non iusto animi accusantium quod velit pariatur nulla, placeat dolores nobis culpa numquam aspernatur consequatur repudiandae commodi? A, itaque minima perferendis animi saepe quod corrupti, ex nostrum atque tenetur neque expedita nisi. Laudantium iure corrupti quaerat voluptates accusamus suscipit aperiam nesciunt, odio non sapiente explicabo magni officiis quam cumque sunt, similique nostrum enim. Eum dolor similique iusto culpa soluta natus qui, ullam est, ab, perspiciatis dolorem omnis? Necessitatibus error voluptas similique placeat sapiente dolorem dolorum nobis culpa eaque amet. Impedit, perferendis dicta.

> Lorem Ipsum
```

Flag: FLAG{COBALT-STRIKE}

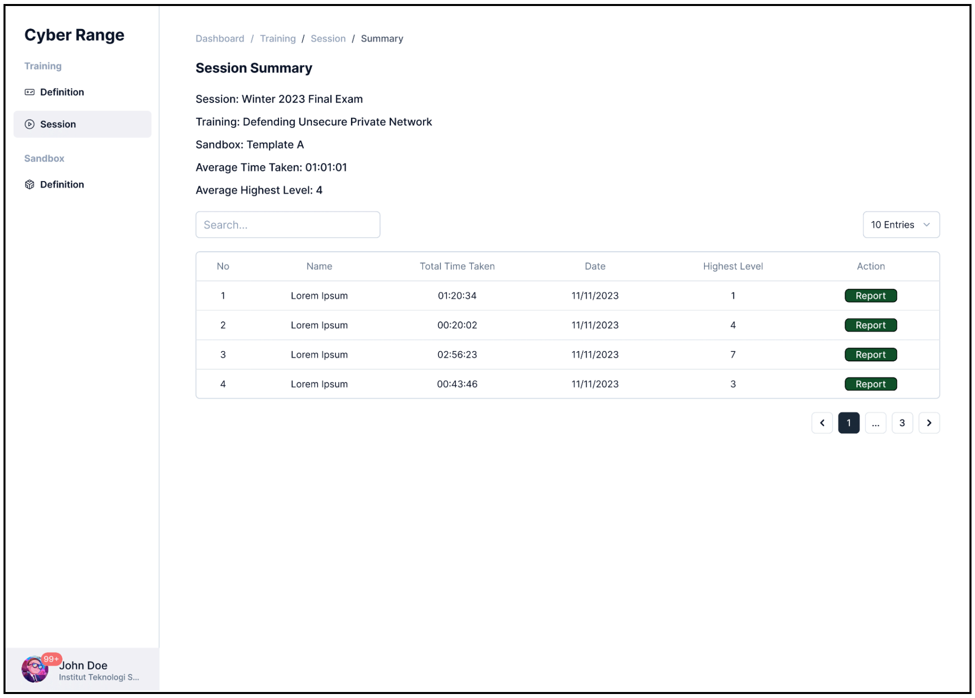
3.7 Adaptive Training Definition



Training Sessions

No	Title	Token	Mode	Start	End	Players	Action
1	Lorem Ipsum	XXX-XXX	Linear	11/11/2023	12/12/2023	123	View Summary
2	Lorem Ipsum	XXX-XXX	Linear	11/11/2023	12/12/2023	509	View Summary
3	Lorem Ipsum	XXX-XXX	Adaptive	11/11/2023	12/12/2023	235	View Summary
4	Lorem Ipsum	XXX-XXX	Adaptive	11/11/2023	12/12/2023	12	View Summary

3.8 Training Sessions



Session Summary

Session: Winter 2023 Final Exam

Training: Defending Unsecure Private Network

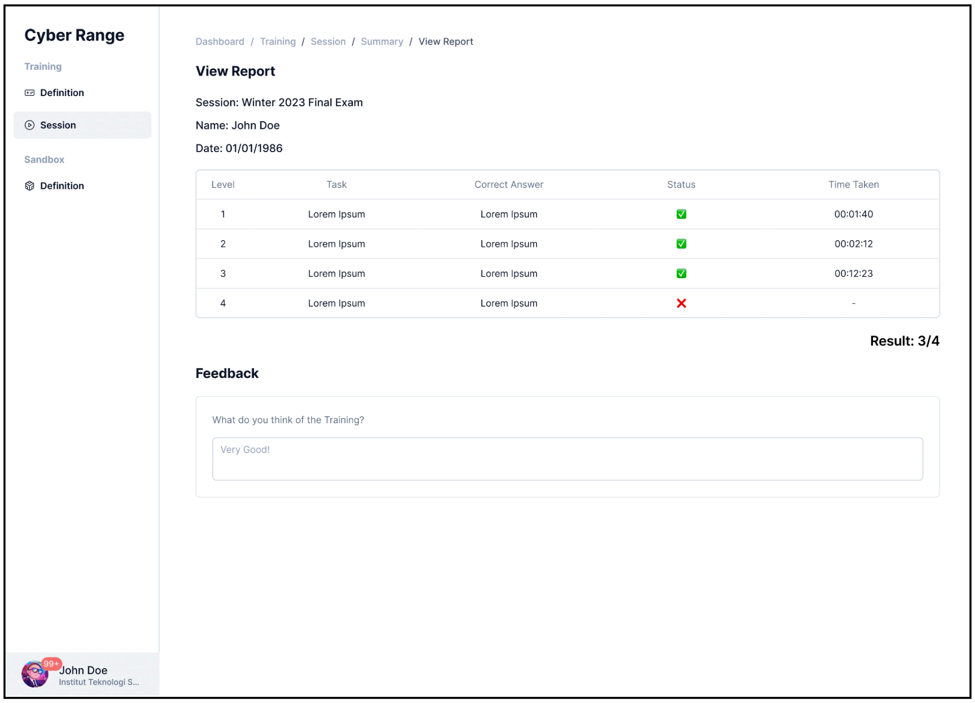
Sandbox: Template A

Average Time Taken: 01:01:01

Average Highest Level: 4

No	Name	Total Time Taken	Date	Highest Level	Action
1	Lorem Ipsum	01:20:34	11/11/2023	1	Report
2	Lorem Ipsum	00:20:02	11/11/2023	4	Report
3	Lorem Ipsum	02:56:23	11/11/2023	7	Report
4	Lorem Ipsum	00:43:46	11/11/2023	3	Report

3.9 Training Summary



View Report

Session: Winter 2023 Final Exam

Name: John Doe

Date: 01/01/1986

Level	Task	Correct Answer	Status	Time Taken
1	Lorem Ipsum	Lorem Ipsum	✓	00:01:40
2	Lorem Ipsum	Lorem Ipsum	✓	00:02:12
3	Lorem Ipsum	Lorem Ipsum	✓	00:12:23
4	Lorem Ipsum	Lorem Ipsum	✗	-

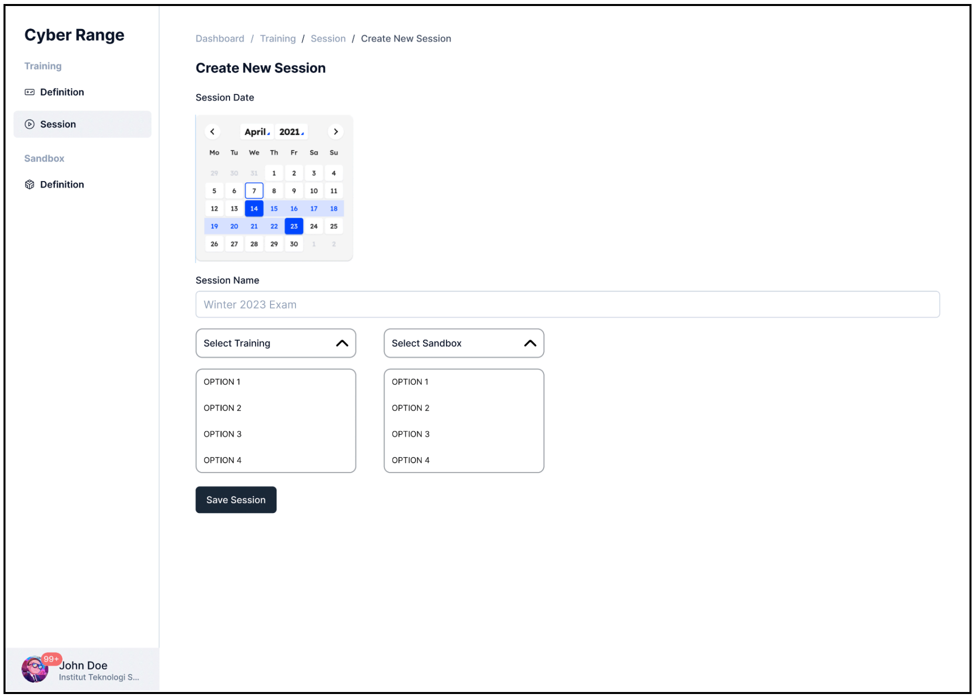
Result: 3/4

Feedback

What do you think of the Training?

Very Good!

3.10 View Report



Cyber Range

Training

Definition

Session

Sandbox

Definition

John Doe
Institut Teknologi S...

Dashboard / Training / Session / Create New Session

Create New Session

Session Date

April 2021

Mo	Tu	We	Th	Fr	Sa	Su
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2

Session Name

Winter 2023 Exam

Select Training

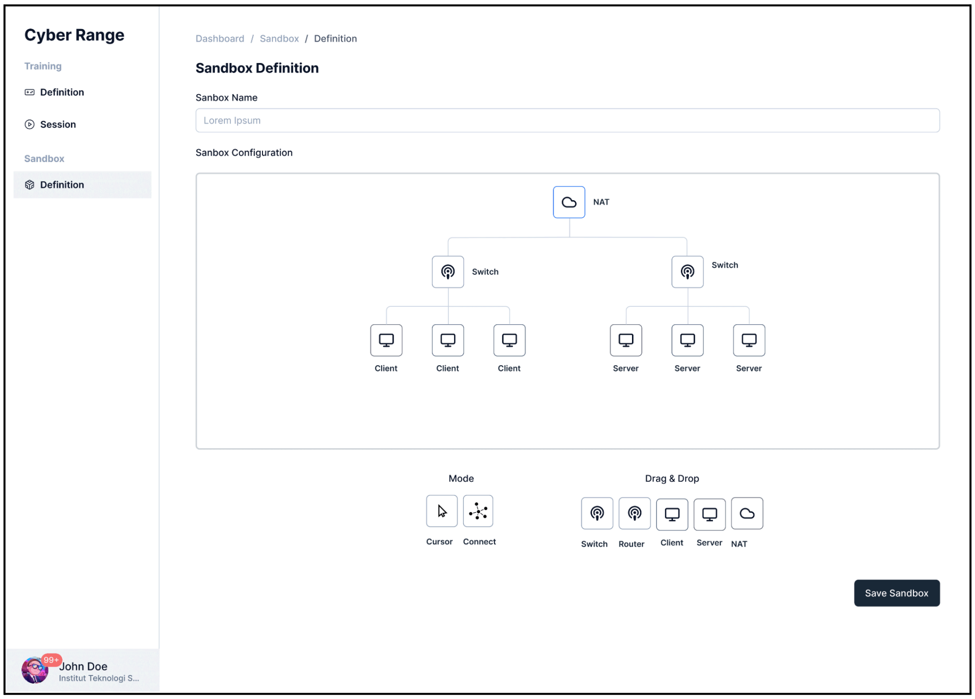
OPTION 1
OPTION 2
OPTION 3
OPTION 4

Select Sandbox

OPTION 1
OPTION 2
OPTION 3
OPTION 4

Save Session

3.11 Create New Session



Cyber Range

Training

Definition

Session

Sandbox

Definition

John Doe
Institut Teknologi S...

Dashboard / Sandbox / Definition

Sandbox Definition

Sanbox Name

Lorem Ipsum

Sanbox Configuration

Mode

Cursor Connect

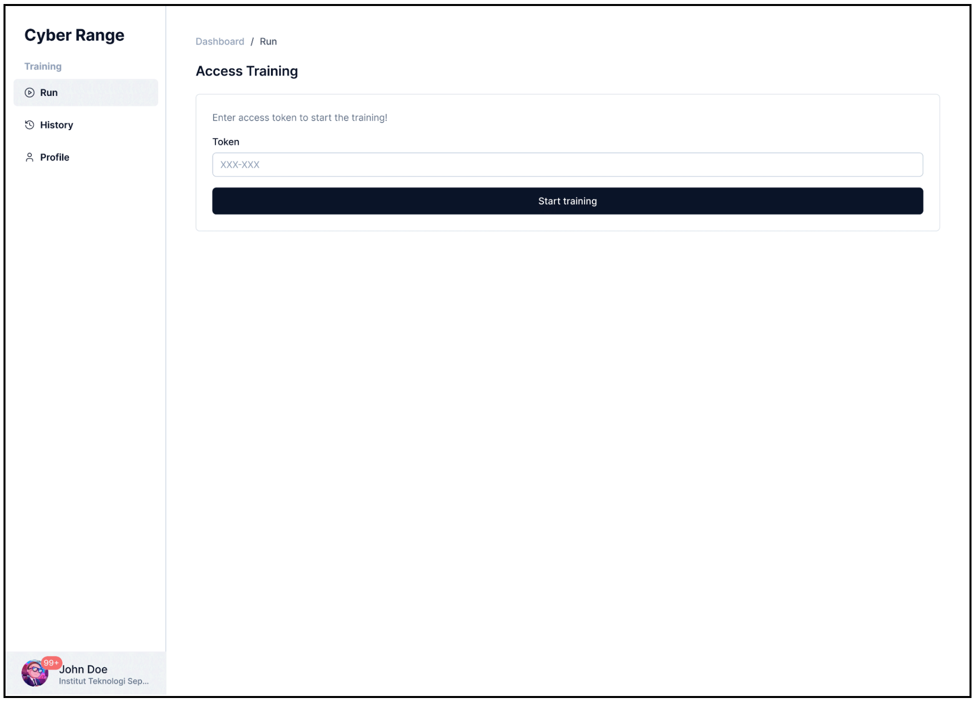
Drag & Drop

Switch Router Client Server NAT

Save Sandbox

3.12 Sandbox Definition

3.1.4 Training User

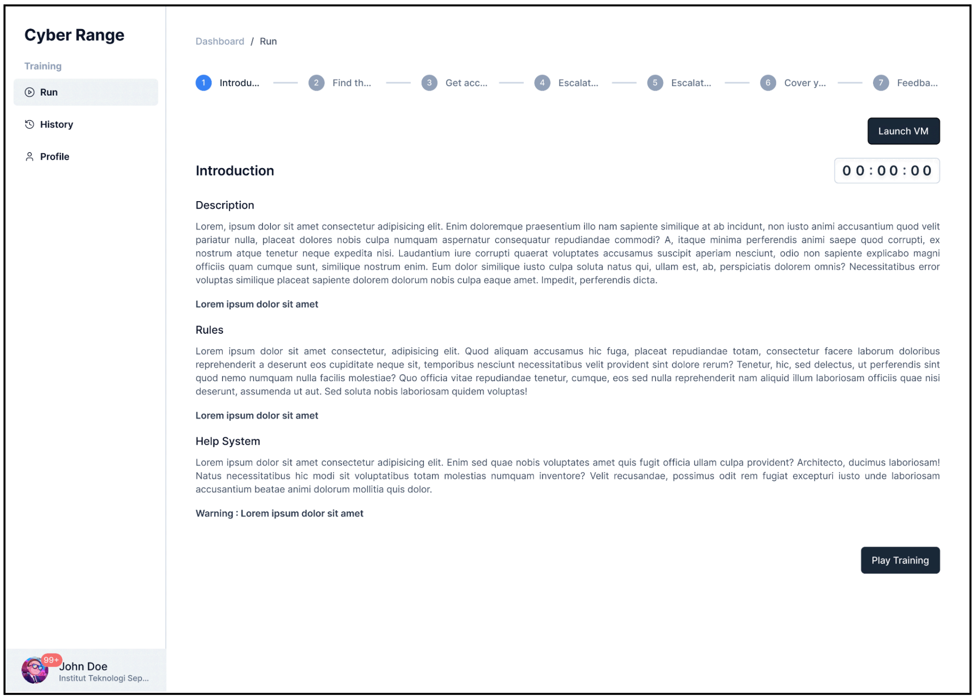


Access Training

Enter access token to start the training!

Token: XXX-XXX

3.13 Access Training



Introduction

Description

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Enim doloremque praesentium illo nam sapiente similique at ab incidunt, non iusto animi accusantium quod velit pariatur nulla, placeat dolores nobis culpa numquam aspernatur consequatur repudiandae commodi? A, itaque minima perferendis animi saepe quod corrupti, ex nostrum atque tenetur neque expedita nisi. Laudantium iure corrupti quaerat voluptates accusamus suscipit aperiam nesciunt, odio non sapiente explicabo magni officiis quam cumque sunt, similique nostrum enim. Eum dolor similique iusto culpa soluta natus qui, ullam est, ab, perspiciatis dolorem omnis? Necessitatibus error voluptas similique placeat sapiente dolorem dolorum nobis culpa eaque amet. Impedit, perferendis dicta.

Lorem ipsum dolor sit amet

Rules

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quod aliquam accusamus hic fuga, placeat repudiandae totam, consectetur facere laborum doloribus reprehenderit a deserunt eos cupiditate neque sit, temporibus nesciunt necessitatibus velit provident sint dolore rerum? Tenetur, hic, sed delectus, ut perferendis sint quod nemo numquam nulla facilis molestiae? Quo officia vitae repudiandae tenetur, cumque, eos sed nulla reprehenderit nam aliquid illum laboriosam officiis quae nisi deserunt, assumenda ut aut. Sed soluta nobis laboriosam quidem voluptas!

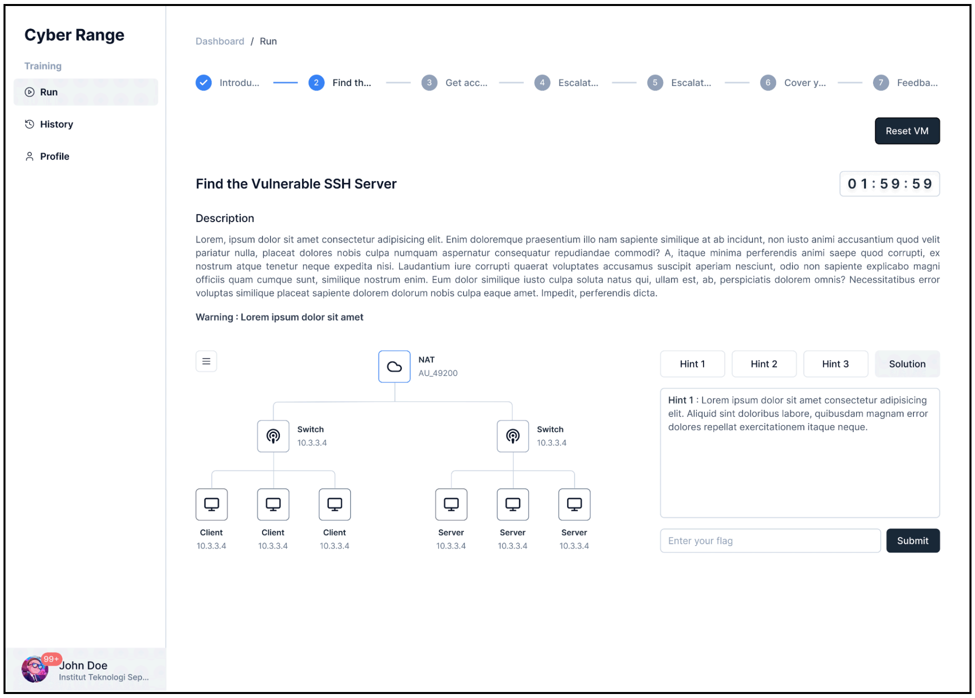
Lorem ipsum dolor sit amet

Help System

Lorem ipsum dolor sit amet consectetur adipisicing elit. Enim sed quae nobis voluptates amet quis fugit officia ullam culpa provident? Architecto, ducimus laboriosam! Natus necessitatibus hic modi sit voluptatibus totam molestias numquam inventore? Velit recusandae, possimus odit rem fugiat excepturi iusto unde laboriosam accusantium beatae animi dolorum mollitia quis dolor.

Warning : Lorem ipsum dolor sit amet

3.14 Play Training (Introduction)



Find the Vulnerable SSH Server

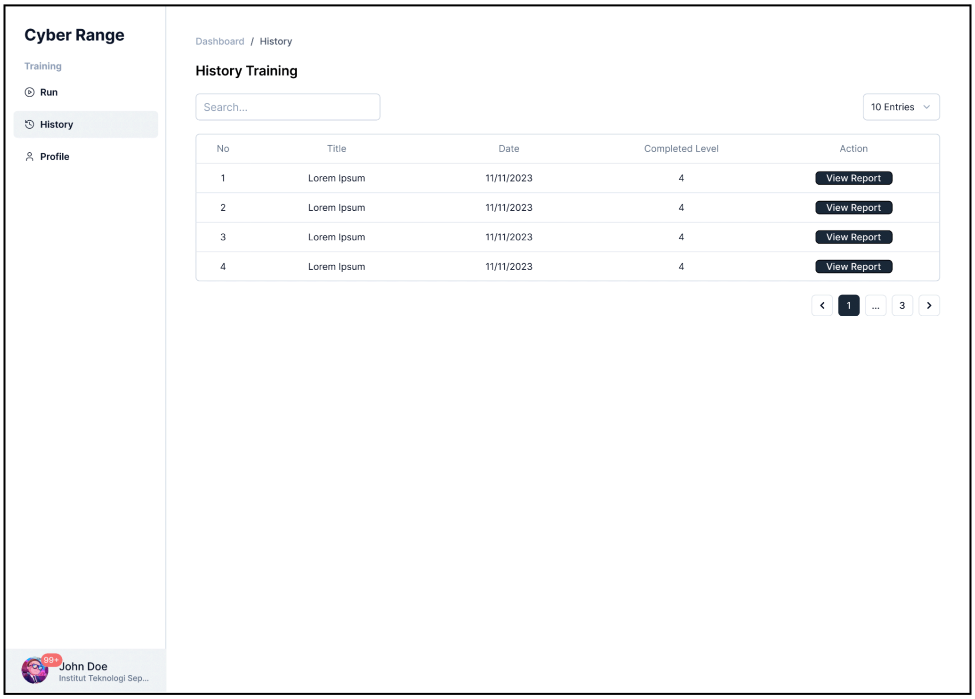
Description

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Enim doloremque praesentium illo nam sapiente similique at ab incidunt, non iusto animi accusantium quod velit pariatur nulla, placeat dolores nobis culpa numquam aspernatur consequatur repudiandae commodi? A, itaque minima perferendis animi saepe quod corrupti, ex nostrum atque tenetur neque expedita nisi. Laudantium iure corrupti quaerat voluptates accusamus suscipit aperiam nesciunt, odio non sapiente explicabo magni officiis quam cumque sunt, similique nostrum enim. Eum dolor similique iusto culpa soluta natus qui, ullam est, ab, perspiciatis dolorem omnis? Necessitatibus error voluptas similique placeat sapiente dolorem dolorum nobis culpa eaque amet. Impedit, perferendis dicta.

Warning : Lorem ipsum dolor sit amet

Hint 1 : Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquid sint doloribus labore, quibusdam magnam error dolores repellat exercitationem itaque neque.

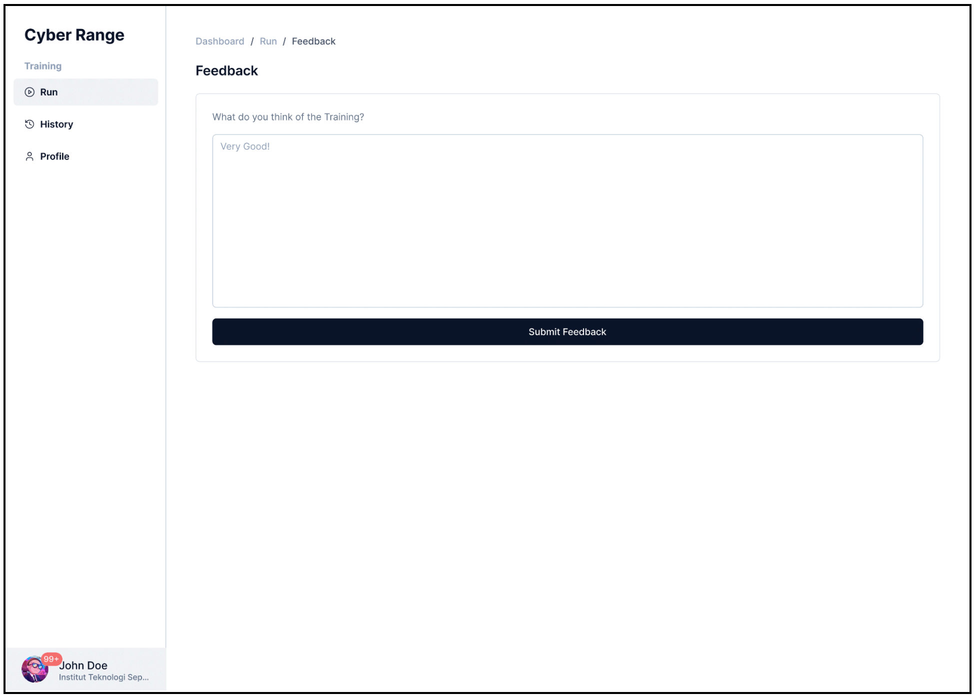
3.15 Play Training



History Training

No	Title	Date	Completed Level	Action
1	Lorem Ipsum	11/11/2023	4	View Report
2	Lorem Ipsum	11/11/2023	4	View Report
3	Lorem Ipsum	11/11/2023	4	View Report
4	Lorem Ipsum	11/11/2023	4	View Report

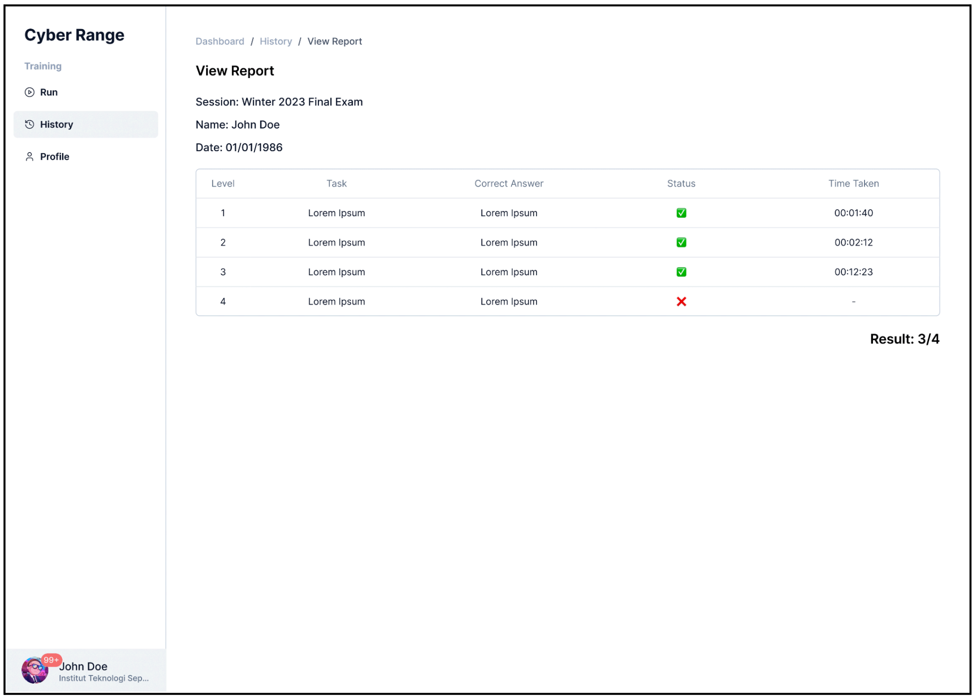
3.16 History Training



What do you think of the Training?

Very Good!

3.17 Feedback



Session: Winter 2023 Final Exam

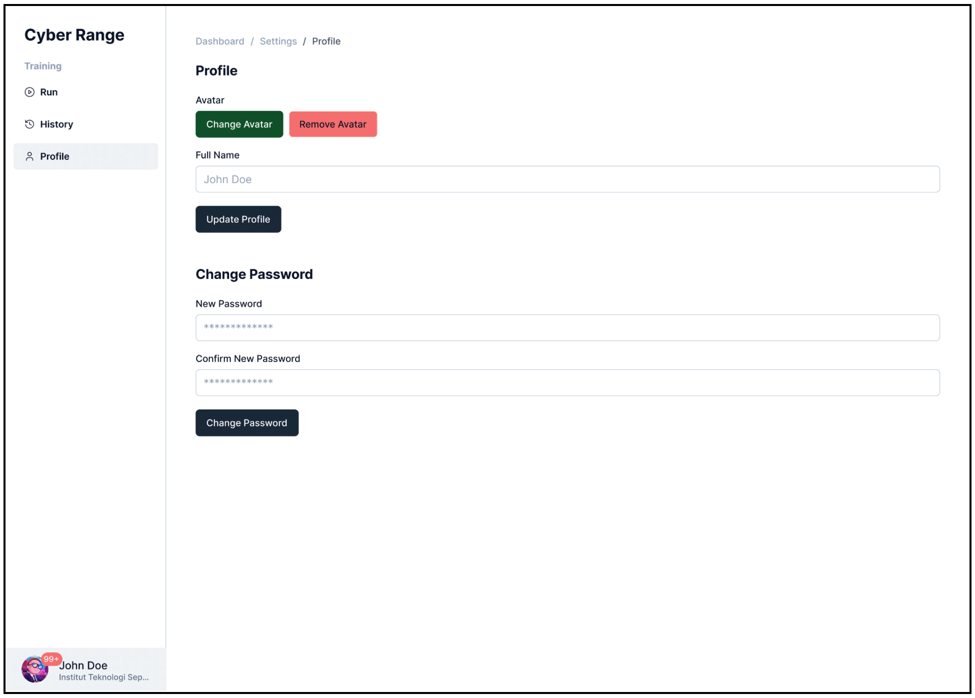
Name: John Doe

Date: 01/01/1986

Level	Task	Correct Answer	Status	Time Taken
1	Lorem Ipsum	Lorem Ipsum	✅	00:01:40
2	Lorem Ipsum	Lorem Ipsum	✅	00:02:12
3	Lorem Ipsum	Lorem Ipsum	✅	00:12:23
4	Lorem Ipsum	Lorem Ipsum	❌	-

Result: 3/4

3.18 View Report



Profile

Avatar

Full Name

John Doe

Change Password

New Password

****

Confirm New Password

****

3.19 View Profile

3.2 Hardware Interfaces

3.2.1 Hypervisor Server (On-Premise)

- ■ Logical Characteristics:
  - Manages virtual environments for realistic threat simulations.
  - Processes Infrastructure as Code (IaC) requests.
- ■ Physical Characteristics:
  - Multicore CPU (e.g., dual hexa-core processors).
  - RAM: ≥ 200GB DDR4.
  - Storage: SSD ≥ 10TB (NVMe recommended for optimal performance).
- ■ Supported Device Types:
  - Compatible with a range of virtualization-ready hardware.
- ■ Communication Protocols:
  - TCP/IP for general communication.
  - SSH for secure virtual machine (VM) access.

3.2.2 Application Server (On-Premise)

- ■ Logical Characteristics:
  - Hosts the database.
  - Processes backend logic.
  - Serves the frontend to end-users.
- ■ Physical Characteristics:
  - Multicore CPU (e.g., quad-core processors).
  - RAM: ≥ 32GB DDR4.

    - Storage: SSD ≥ 512GB (NVMe recommended for optimal database performance).
- ■ Supported Device Types:
    - Compatible with standard server hardware.
- ■ Communication Protocols:
    - TCP/IP for general communication.
    - HTTP/HTTPS for web-based communication.
    - JDBC/ODBC for database connectivity.

3.3 Software Interfaces

3.3.1 Database: PostgreSQL

- ■ Connection
    - The application interfaces with a PostgreSQL database (version 16.0) to store and retrieve dynamic data.
- ■ Data Items:
    - User information, including authentication details.
    - Training tasks and session metadata.
    - Configuration settings for sandboxes.
- ■ Purpose:
    - User data is stored for authentication and personalized experiences.
    - Training-related information is managed for tracking and reporting purposes.
    - Configuration settings are stored for sandbox setup and customization.

3.3.2 Backend: Golang

- ■ Connection
    - The backend, developed in Golang (version 1.21.1), facilitates communication between the frontend and the database.
- ■ Data Items:
    - Receives user requests and processes them.
    - Manages infrastructure requests and interactions with the Hypervisor.
- ■ Services:
    - Exposes RESTful APIs for frontend communication.
    - Utilizes PostgreSQL drivers for database interactions.
- ■ Communication:
    - HTTP/HTTPS for frontend-to-backend communication.
    - Database queries and transactions for data retrieval and storage.

3.3.3 Frontend: NextJS

- ■ Connection:
    - The frontend, built with NextJS, provides an interactive user interface.
- ■ Data Items:
    - User input and requests.

        - Displayed data received from the backend.
    - Services:
        - Consumes RESTful APIs provided by the backend.
    - Communication:
        - HTTP/HTTPS for frontend-to-backend communication.

3.3.4 Hypervisor: Proxmox

Configures virtual environments and virtual machines based on predefined specifications.

- Connection:
    - The application interfaces with Proxmox (version X.X) for virtual environment configuration.
- Data Items:
    - Virtual environment specifications.
    - Commands for virtual machine orchestration.
- Services:
    - Utilizes Proxmox API for managing virtual environments.
- Communication:
    - Proxmox API protocols for sending commands and receiving status updates.

3.4 Communications Interfaces

3.4.1 User Access and VM Interaction

- Access Method : Users can access virtual machines using SSH once connected to the VPN.
- SSH Authentication : username and password based authentication is enforced for ease of use.
- IP Addressing : Internal IP addresses assigned to the VMs within the local network.
- Port Number : SSH port for VM access.

3.4.2 API Communication: RESTful API

- Data Formats Accepted : The RESTful API accepts both JSON and URL-encoded form data for communication between the frontend and backend. This ensures flexibility in data transmission based on different use cases or client preferences.

3.4.3 Terraform Integration

- Integration Method : Terraform scripts are generated and managed by the Golang backend.
- Configuration Storage : Terraform configurations are stored as files on the Application Server.
- Flow : The Golang backend, upon receiving infrastructure requests, generates Terraform

configurations based on predefined specifications. These configurations are then stored on the Application Server. The backend initiates the Terraform execution, deploying and configuring virtual environments on either the Hypervisor Server (On-Premise).

4. System Features

4.1 Account Registration

REQ-01	Register new user	
Priority	Medium	
Actor	Admin	
Pre-condition	Training User and/or Training Maker hasn't previously registered and actor is currently in the Create New User page	
Post-condition	New Users data is stored in the database and each can be used for authentication purposes respectively to access the system.	
Normal Flow	***Actor***	***System***
		1. Presents a form for registering a new account and a button for uploading a CSV file.
	2. Completes the form for a new user.	
		3. Processes the information and performs validation.
		4. Upon successful validation, the Actor data is stored in the

		database.
Alternative Flow	2.1. Uploads a CSV file containing a list of user information formatted according to the specifications outlined in the assumption section.	
Exception Flow		3.1. Validation fails, the system displays an error message specifying the nature of the error for the administrator to address.
		3.2. CSV is in the incorrect format, the system presents an error message detailing the specific issue, guiding the administrator on the necessary corrections.

4.2 Account Login

REQ-04	User Login
Priority	High
Actor	Admin, Training User, Trainer User
Pre-condition	Actors are unable to access the platform and is currently in the login page
Post-condition	Actors gain access to the platform

Normal Flow	***Actor***	***System***
	1. Enter credentials.	
		2. Validates the provided credentials.
		3. Upon successful validation, actor receive an access token and the system redirects the user to their dashboard based on their assigned role.
Exception Flow		2.1. Validation fails, the system displays an error message indicating that the provided credentials are incorrect.

4.3 Update User Profile

REQ-02	Update profile	
Priority	Low	
Actor	Training User	
Pre-condition	Actor are registered and is currently on the profile setting page	
Post-condition	Actor informations are updated	
Normal Flow	***Actor***	***System***
		1. Presents a form pre-filled with the current actor information.

	2. Modifies the information as desired.	
		3. Validates the entered information.
		4. Upon successful validation, the modified user information is stored in the database.
Exception Flow		3.1. Validation fails, the system displays an error message.

4.4 Account Deletion

REQ-03	Delete user	
Priority	Low	
Actor	Admin	
Pre-condition	The corresponding Training User(s) are registered and actor is currently in the User List page	
Post-condition	The corresponding Training User(s) is removed from the platform and it's information is deleted from the database	
Normal Flow	***Actor***	***System***
		1. Presents a list of registered training users and/or training makers.
	2. Selects one or more users for deletion.	

		3. Displays a pop-up containing the selected users and requests confirmation for deletion.
	4. Confirms the deletion	
		5. Prompts the admin to enter their password.
	6. Provides the password.	
		7. Validates the provided password.
		8. Upon successful validation, the selected user(s) are removed, and their information is deleted from the database.
Alternative Flow	4.1. Denies the confirmation, the use case ends.	
Exception Flow		7.1. Password validation fails, take no action, and the use case ends.

4.5 Sandbox Definition

REQ-05	Define sandbox environment
Priority	High
Actor	Training Maker

Pre-condition	Actor is currently in the Sandbox Definition page	
Post-condition	A new Sandbox definition is created	
Normal Flow	***Actor***	***System***
	1. Completes the form by providing a name for the sandbox.	
	2. Utilizes the drag & drop frontend utility to configure the sandbox topology, connecting various components such as VMs, switches, routers, and other relevant elements.	
	3. Upon completion, click the "Save Sandbox" button.	
		4. Processes and converts the received data into a Terraform configuration file, which is then stored both on the server and in the database. This file encapsulates the specified sandbox settings and ensures seamless retrieval and application for future use.
Exception Flow		4.2. System fails to parse the given data into a Terraform configuration file and returns an error message.

4.6 Training Tasks Definition

REQ-06	Define training tasks	
Priority	High	
Actor	Training Maker	
Pre-condition	Actor is currently in the Training Definition page	
Post-condition	A new Training is successfully created.	
Normal Flow	***Actor***	***System***
		1. Displays the page, defaulting to the "linear mode" path.
	2. For each training task, Actor completes a form that includes a description, which can be formatted in Markdown, hints if necessary, and the flag required to pass the task.	
	3. For each training task, the actor completes a form that includes a description, which can be formatted in Markdown, hints if necessary, and the flag required to pass the task.	
	4. The actor clicks the "Save Task" button.	
		5. Saves the tasks and the training definition in the database

	6. Repeat steps 3 and 4 for as many tasks as desired.	
		7. System saves the tasks in the database.
Alternative Flow	3.1. The actor then clicks the "Switch to Adaptive Mode." 3.2. In Adaptive Mode, the actor fills out a similar form for each task as in linear mode, but with an additional conditional form. This form defines the conditions that need to be met in case the task involves branching.	

4.7 Define Training Session

REQ-07	Define training session	
Priority	High	
Actor	Training Maker	
Pre-condition	Actor is currently in the Training Session dashboard	
Post-condition	A new Training Session Token is successfully created.	
Normal Flow	***Actor***	***System***
	1. Initiates the session creation by clicking the "Create New	

	Session" button.	
	2. Proceeds to fill out the form, which includes specifying the start and end dates of the session, providing a session name, and selecting the training and sandbox upon which the session is based.	
	3. Upon completing the form, the actor clicks the "Save Session" button.	
		4. The system generates a unique token for the session and saves the details in the database.
Exception Flow		2.1. The provided start date and/or end date is in the past, the system will refuse to save the details until the configuration is corrected.

4.8 Play Training Session

REQ-08	Play training
REQ-09	Generate training report
REQ-10	Give feedback
Priority	High

Actor	Training User	
Pre-condition	The Actor is currently in the Access Training page	
Post-condition	The Training user successfully completes their training session, and the system generates a report of their engagement.	
Normal Flow	***Actor***	***System***
	1. Enter a session token.	
		2. Validates the token and retrieves the corresponding training details.
		3. Presents the training page to the actor.
	4. Click the "Launch VM" button.	
		5. Initiates the boot-up process, launching the configured topology components, and returns the active duration to the page.
	6. Click the "Play Training" button.	
		7. Presents the training tasks.
	8. Diligently completes the required tasks.	
	9. Upon task completion, the actor submits the flag received as a proof for accomplishing the	

	task.	
		10. Validates the submitted flag.
		11. If the validation is successful, the system proceeds to the next task page.
	12. Repeat step 8 until all tasks are successfully completed.	
		13. Generates a report for the engagement and saves the result in the database.
		14. Fills the optional (can be left as blank) Feedback form and click the submit button
Alternative Flow	8.1. The actor may click the "Hint/Solution" button.	8.2. Reveals the hint/solution to assist the actor in understanding and completing the task effectively.
		10.1. Validation fails, the actor is required to repeat step 8, providing an opportunity for correction and ensuring the accuracy and integrity of the training process.

4.9 Reset Sandbox

REQ-12	Reset sandbox

Priority	Medium	
Actor	Training User	
Pre-condition	Sandbox is launched and is currently in the Training page	
Post-condition	Sandbox is successfully destroyed and re-launched	
Normal Flow	***Actor***	***System***
	1. Clicks the “Reset VM” button	
		2. Destroy the associated sandbox.Conducts a cleanup process to remove any residual components or data associated with the terminated sandbox.
		3. Initiates the launch of a new sandbox to replace the previous one.
		4. System returns and updates the relevant information needed for display to the actor.
Exception Flow		2.1. Communication with the Hypervisor server fails, notifies the user to contact the administrator.

4.10 Access Sandbox

REQ-11	Access sandbox environment

Priority	High	
Actor	Training User	
Pre-condition	Sandbox is launched and is in the Training page	
Post-condition	Training User gain access to the Sandbox	
Normal Flow	***Actor***	***System***
	1. Establishes a connection to the internal VPN.	
	2. Connects to the virtual machine using the SSH protocol, utilizing the provided IP and other relevant information available on the training page.	
		3. Validates the provided credentials.
	4. Gains access to the sandbox and proceeds to execute various tasks.	
Alternative Flow		3.1. Validation fails, redirect back to step 2
Exception Flow		2.1. Communication fails, flow diverts to the "Reset VM" use case

4.11 Force Stop Sandbox

REQ-13	Stop sandbox	
Priority	High	
Actor	Admin	
Pre-condition	The Sandbox is launched and is currently in the training instance's dashboard.	
Post-condition	The Sandbox is successfully destroyed.	
Normal Flow	***Actor***	***System***
		1. Displays a list of running sandboxes.
	2. Click the "Stop" button next to the relevant sandbox.	
		3. Destroy the associated sandbox.
		4. Conducts a cleanup process to remove any residual components or data associated with the terminated sandbox.

4.12 Force Reset Sandbox

REQ-12	Reset sandbox
Priority	High

Actor	Admin	
Pre-condition	The Sandbox is launched and is currently in the training instance's dashboard.	
Post-condition	Sandbox is successfully destroyed and re-launched	
Normal Flow	***Actor***	***System***
		1. Displays a list of running sandboxes.
	2. Click the "Reset" button next to the relevant sandbox.	
		3. Destroy the associated sandbox.
		4. Conducts a cleanup process to remove any residual components or data associated with the terminated sandbox.
		5. Initiates the launch of a new sandbox to replace the previous one.
		6. System returns and updates the relevant information needed for display to the Training User.
Exception Flow		2.1. Communication with the Hypervisor server fails, displays the error message.

4.13 View Training Report

REQ-14	View training report	
Priority	Medium	
Actor	Training User	
Pre-condition	Actor is currently in the Training History dashboard.	
Post-condition	Actor able to view their Training Report	
Normal Flow	***Actor***	***System***
		1. Displays the actor's list of participated training.
	2. Click the "View Report" button	
		3. Display the report for the associated training

4.14 View Training Summary

REQ-14	View training report
REQ-15	View feedback
Priority	High
Actor	Training Maker

Pre-condition	Actor is currently in the Training Session dashboard.	
Post-condition	Actors are able to view the summary of the Training session as well as the report and feedback of each training user who participated.	
Normal Flow	***Actor***	***System***
		1. Displays a list of sessions.
	2. Click the “View Summary” button	
		3. Display the summary for the associated session and list of participated Training Users.
	4. Click the “Report” button	
		5. Display the report and feedback for the associated Training User

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Response Time

- Cyber Range should have a maximum response of 700ms for user interactions (e.g., login and starting a training session) under normal operating conditions.
- The system should maintain a consistent response time of 500ms during peak usage periods.

5.1.2 Scalability

- The system must be able to scale to accommodate at least 40 concurrent users.
- The time required to provision a new sandbox environment should not exceed 2 hours

5.2 Safety Requirements

5.2.1 User Data Protection

- The platform must comply with data protection regulations, ensuring the confidentiality and integrity of user data.
- In the event of a system failure, there should be mechanisms in place to prevent data loss or corruption.

5.2.2 User Notification

- In case of system maintenance or unexpected downtime, users must be notified in advance.

5.3 Security Requirements

5.3.1 Authentication

- User authentication must be performed securely, utilizing strong encryption algorithms.

5.3.2 Data Encryption

- All data transmission between the client and server must be encrypted using industry-standard protocols.
- Sensitive user data stored in the database must be encrypted, and access should be restricted based on roles.

5.3.3 Access Control

- Role-based access control (RBAC) should be enforced, restricting user access based on their roles.
- The platform must log and monitor access attempts, providing an audit trail for security analysis.

5.4 Software Quality Attributes

5.4.1 Maintainability

- Code should follow best practices, be well-documented, and adhere to a modular structure for ease of maintenance.
- Regular code reviews and automated testing should be conducted to ensure maintainability.

5.4.2 Usability

- The user interface must be intuitive, with clear navigation and informative feedback.

- The platform should provide user guides and documentation to facilitate ease of use.

5.4.3 Reliability

- The system should have an uptime of at least 80%
- Automated backup mechanisms must be in place to prevent data loss in the event of system failures.

5.5 Business Rules

5.5.1 User Registration

- Only administrators have the authority to register new users on the platform.

5.5.2 Sandbox Environment Management

- Instructors have the authority to define and modify sandbox environments for training sessions.

5.5.3 Training Session Feedback

- Users are able to provide feedback after completing a training session.
- Feedback should be constructive and relate to the effectiveness of the training content.

5.5.4 Training Task Definition

- Instructors are responsible for defining training tasks and scenarios for users.
- Training tasks cannot be edited or deleted once assigned to a training session for synchronization reasons.

6. Other Requirements

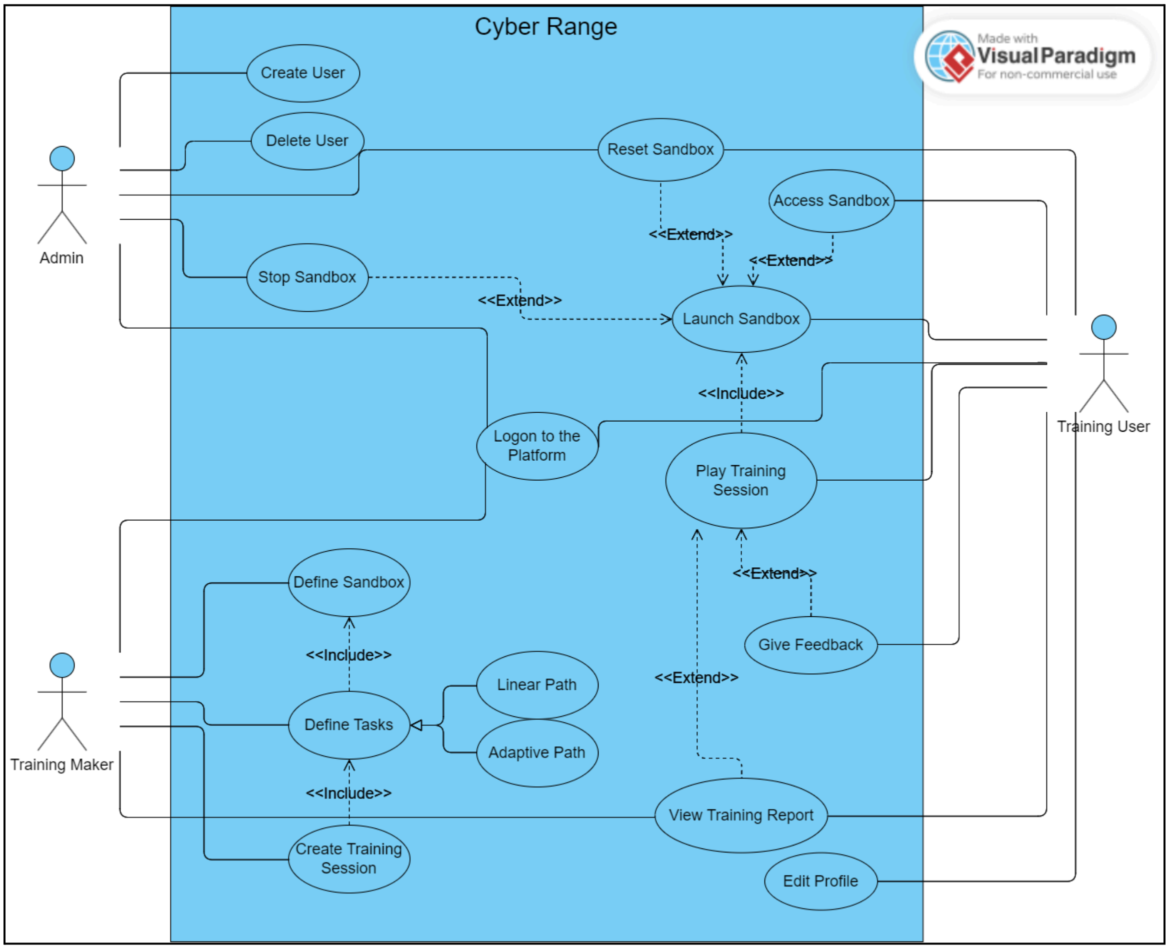
Appendix A: Glossary

Below are definitions or explanations for specific terms, acronyms, or concepts. This section serves as a reference for readers who may encounter unfamiliar terminology.

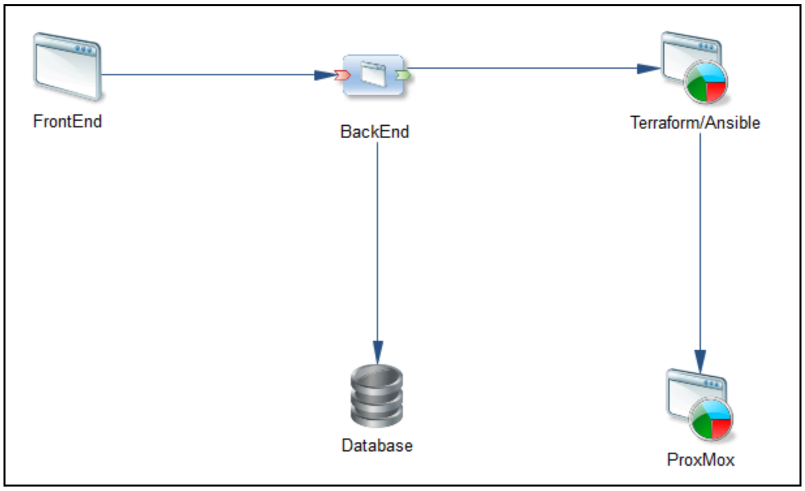
Terms	*Definition*
Sandbox	A controlled and isolated environment that allows users to simulate real-life scenarios of which the training is based upon. Also synonym with Virtual Environment and Simulated Environment.

Topology	The arrangement or configuration of a system's components, including the relationships and connections between them.
Training	Training an umbrella that consists of Training Tasks, Session and Report.
Training Tasks	Specific activities or assignments that need to be completed in order to complete the Training.
Training Session	A scheduled period of time where the Training could be accessed.
Training Report	A summary of the outcomes and progress used for assessment and improvement purposes.
Virtual Machine (VM)	A software emulation of a physical computer that runs in an isolated environment, allowing multiple operating systems to coexist on the same hardware.
Hypervisor	A software layer that enables multiple operating systems to run on a single physical host, managing and allocating resources among virtual machines.

Appendix B: Analysis Models



6.1 Use Case Diagram



6.2 Architecture Diagram

Appendix C: To Be Determined List

1. *Libraries to be used in the development of the backend and frontend server.*
2. *VM templates for Proxmox to use when deploying the sandbox.*
3. *Terraform and Proxmox global configuration.*
4. *Sandbox Definition data translation between backend and frontend for backend to parse into Terraform configuration files.*